Trabajo practico 2

Tareas sobre el dataset Breast Cancer Winsconsin.

```
library(class)
library(gmodels)
# Leyendo dataset
data <- read.csv("http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc
# Ignorar ids
data <- data[-1]
# funcion de normalizacion
normalize <- function(x) {
  return ((x-min(x))/(max(x)-min(x)))
}

data_n <- as.data.frame(lapply(data[2:31], normalize))

data_train <- data_n[1:469, ]
data_test  <- data_n[470:569, ]

data_train_labels <- data[1:469, 1]
data_test_labels  <- data[470:569, 1]

data_test_pred <- knn(train=data_train, test=data_test, cl=data_train_labels, k=21)
CrossTable(x=data_test_labels, y=data_test_pred, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                  | data_test_pred
## data_test_labels |         B |         M | Row Total |
## -----------------|-----------|-----------|-----------|
##                B |        77 |         0 |        77 |
##                  |     1.000 |     0.000 |     0.770 |
##                  |     0.975 |     0.000 |           |
##                  |     0.770 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##                M |         2 |        21 |        23 |
##                  |     0.087 |     0.913 |     0.230 |
##                  |     0.025 |     1.000 |           |
##                  |     0.020 |     0.210 |           |
```

```
## -----------------|-----------|-----------|-----------|
##     Column Total |        79 |        21 |       100 |
##                  |     0.790 |     0.210 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

Ejercicios.

Mejore el rendimiento utilizando una normalizacion con z-scores provista por la funcion scale() de R.

```
data_n_z <- as.data.frame(scale(data[2:31]))
data_train_z <- data_n_z[1:469, ]
data_test_z  <- data_n_z[470:569, ]
data_train_labels_z <- data[1:469, 1]
data_test_labels_z  <- data[470:569, 1]
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=21)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                    | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        77 |         0 |        77 |
##                    |     1.000 |     0.000 |     0.770 |
##                    |     0.975 |     0.000 |           |
##                    |     0.770 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         2 |        21 |        23 |
##                    |     0.087 |     0.913 |     0.230 |
##                    |     0.025 |     1.000 |           |
##                    |     0.020 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        79 |        21 |       100 |
##                    |     0.790 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

Pruebe algunos valores alternativos de k=1, 5, 11, 15, 21 y seleccione el mejor valor de k.

```
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=1)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                    | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        73 |         4 |        77 |
##                    |     0.948 |     0.052 |     0.770 |
##                    |     0.973 |     0.160 |           |
##                    |     0.730 |     0.040 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         2 |        21 |        23 |
##                    |     0.087 |     0.913 |     0.230 |
##                    |     0.027 |     0.840 |           |
##                    |     0.020 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        75 |        25 |       100 |
##                    |     0.750 |     0.250 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=5)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
```

```
##
##
##                    | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        73 |         4 |        77 |
##                    |     0.948 |     0.052 |     0.770 |
##                    |     1.000 |     0.148 |           |
##                    |     0.730 |     0.040 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         0 |        23 |        23 |
##                    |     0.000 |     1.000 |     0.230 |
##                    |     0.000 |     0.852 |           |
##                    |     0.000 |     0.230 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        73 |        27 |       100 |
##                    |     0.730 |     0.270 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```r
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=11)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                    | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        76 |         1 |        77 |
##                    |     0.987 |     0.013 |     0.770 |
##                    |     0.987 |     0.043 |           |
##                    |     0.760 |     0.010 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         1 |        22 |        23 |
##                    |     0.043 |     0.957 |     0.230 |
##                    |     0.013 |     0.957 |           |
##                    |     0.010 |     0.220 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        77 |        23 |       100 |
##                    |     0.770 |     0.230 |           |
## -------------------|-----------|-----------|-----------|
##
```

```
##
```

```
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=15)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
##                     | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        77 |         0 |        77 |
##                    |     1.000 |     0.000 |     0.770 |
##                    |     0.975 |     0.000 |           |
##                    |     0.770 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         2 |        21 |        23 |
##                    |     0.087 |     0.913 |     0.230 |
##                    |     0.025 |     1.000 |           |
##                    |     0.020 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        79 |        21 |       100 |
##                    |     0.790 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

```
data_test_pred_z <- knn(train=data_train_z, test=data_test_z, cl=data_train_labels, k=21)
CrossTable(x=data_test_labels_z, y=data_test_pred_z, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  100
##
##
```

```
##                    | data_test_pred_z
## data_test_labels_z |         B |         M | Row Total |
## -------------------|-----------|-----------|-----------|
##                  B |        77 |         0 |        77 |
##                    |     1.000 |     0.000 |     0.770 |
##                    |     0.975 |     0.000 |           |
##                    |     0.770 |     0.000 |           |
## -------------------|-----------|-----------|-----------|
##                  M |         2 |        21 |        23 |
##                    |     0.087 |     0.913 |     0.230 |
##                    |     0.025 |     1.000 |           |
##                    |     0.020 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##       Column Total |        79 |        21 |       100 |
##                    |     0.790 |     0.210 |           |
## -------------------|-----------|-----------|-----------|
##
##
```

- mientras termina su merecido cafe verifique si el resultado cambia utilizando pacientes elegidos aleatoriamente para el conjunto de validacion.

```
normalize_cols <- function(dataf, method='mm', cols=NULL) {
  # select all columns if none provided
  if (is.null(cols)) {
    cols <- seq_len(ncol(dataf))
  }
  switch(method,
    'mm'={norm_fun = normalize},
    'zs'={norm_fun = scale}
  )
  dataf[,cols] <- as.data.frame(lapply(dataf[,cols], norm_fun))
  return (dataf)
}
split_train_test <- function(data, percentage=.80, seed=1) {
  smp_size <- percentage*nrow(data)
  set.seed(1)
  sample <- sample.int(n = nrow(data), size = floor(smp_size), replace = F)
  train <- data[sample, ]
  test  <- data[-sample, ]
  return (list('train'=train, 'test'=test, 'sample_idx'=sample))
}

new_data <- normalize_cols(data, method="zs", cols=2:31)
new_data <- split_train_test(new_data, percentage=.80)
data_train <- new_data$train[,2:31]
data_test <- new_data$test[,2:31]
data_train_labels <- new_data$train[,1]
data_test_labels <- new_data$test[,1]
# data_n <- as.data.frame(lapply(data[2:31], normalize))

data_test_pred <- knn(train=data_train, test=data_test, cl=data_train_labels, k=21)
CrossTable(x=data_test_labels, y=data_test_pred, prop.chisq = FALSE)
```

```
##
```

```
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  114
##
##
##                  | data_test_pred
## data_test_labels |         B |         M | Row Total |
## -----------------|-----------|-----------|-----------|
##                B |        66 |         0 |        66 |
##                  |     1.000 |     0.000 |     0.579 |
##                  |     0.917 |     0.000 |           |
##                  |     0.579 |     0.000 |           |
## -----------------|-----------|-----------|-----------|
##                M |         6 |        42 |        48 |
##                  |     0.125 |     0.875 |     0.421 |
##                  |     0.083 |     1.000 |           |
##                  |     0.053 |     0.368 |           |
## -----------------|-----------|-----------|-----------|
##     Column Total |        72 |        42 |       114 |
##                  |     0.632 |     0.368 |           |
## -----------------|-----------|-----------|-----------|
##
##
```

Practico 2

En este trabajo se hizo un procesamiento sobre uno de los datasets de kaggle. Se eligió un dataset sobre consumo de alcohol de estudiantes. El dataset está disponible en este link.

Dados estos datos se trató de predecir el consumo de alcohol dadas las variables disponibles en el dataset. Se trató solamente las variables numéricas del dataset con el algoritmo K Nearest Neighbours (KNN).

```
data_alc <- read.table("../input/student-mat.csv",sep=",",header=TRUE)
```

Deshechamos variables no numéricas y cambiamos las etiquetas de la variables que nos interesa procesar, empleando nombres más significativos.

```
# get numeric columns to work with
num_cols <- unlist(lapply(data_alc, is.numeric))
data_alc <- data_alc[, num_cols] # forget other variables
# anotate correctly our labels
data_alc$Walc <- gsub(1, 'Very-Low', data_alc$Walc)
data_alc$Walc <- gsub(2, 'Med-Low', data_alc$Walc)
data_alc$Walc <- gsub(3, 'Med', data_alc$Walc)
data_alc$Walc <- gsub(4, 'Med-High', data_alc$Walc)
data_alc$Walc <- gsub(5, 'Very-High', data_alc$Walc)
```

A continuación se probaran diferentes valores del parámetro K, discriminando por la forma de normalizar el dataset.

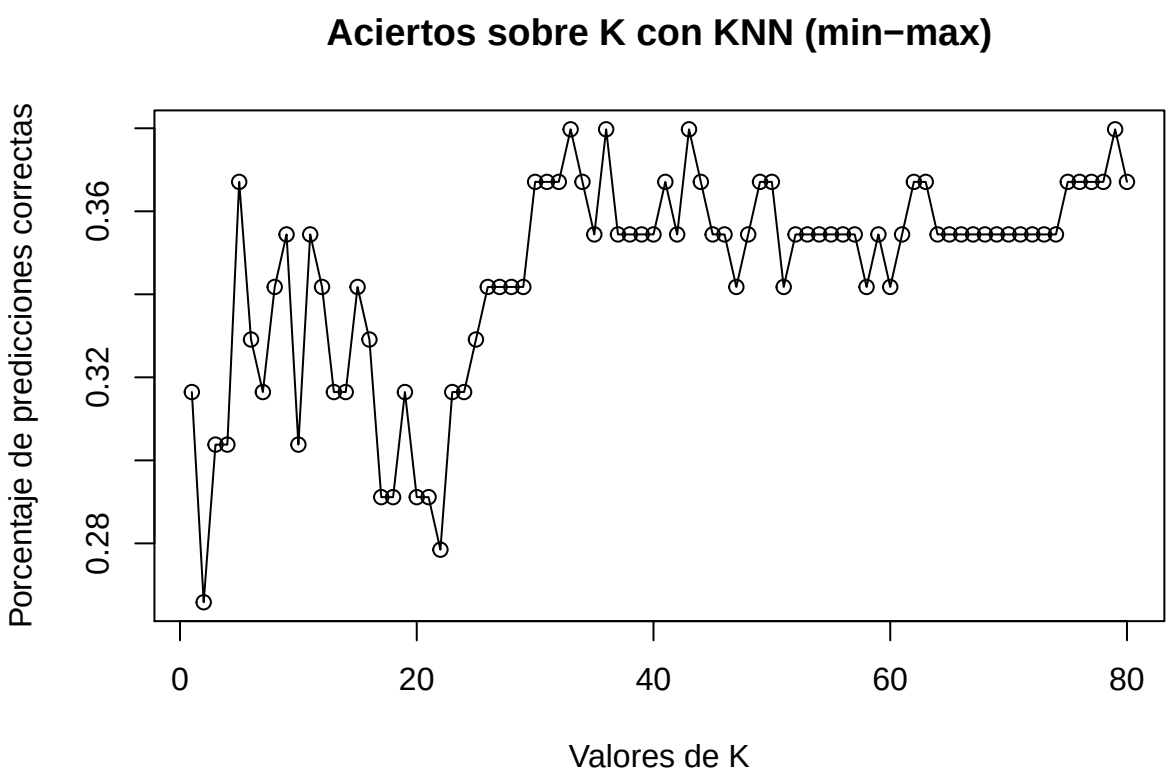
Procesamiento con Normalización Min-Max

```
cols_to_norm <- setdiff(1:16, 11) # columns to normalize

# Min max normalization
new_data <- normalize_cols(data_alc, method="mm", cols=cols_to_norm)
splits_tt <- split_train_test(new_data, percentage=.80, seed=1)
data_train <- splits_tt$train[,cols_to_norm]
data_test <- splits_tt$test[,cols_to_norm]
data_train_labels <- splits_tt$train[,11]
data_test_labels <- splits_tt$test[,11]


accuracy_for_k <- function(k) {
  data_test_pred <- knn(train=data_train, test=data_test, cl=data_train_labels, k=k)
  # res = CrossTable(x=data_test_labels, y=data_test_pred, prop.chisq = FALSE)
  # accuracy calculation
  accuracy = sum(data_test_pred == data_test_labels) / length(data_test_labels)
  return(accuracy)
}

acs_mm <- invisible(lapply(1:80, accuracy_for_k))
plot(1:length(acs_mm), acs_mm, type='o', xlab='Valores de K', ylab='Porcentaje de predicciones correctas
```



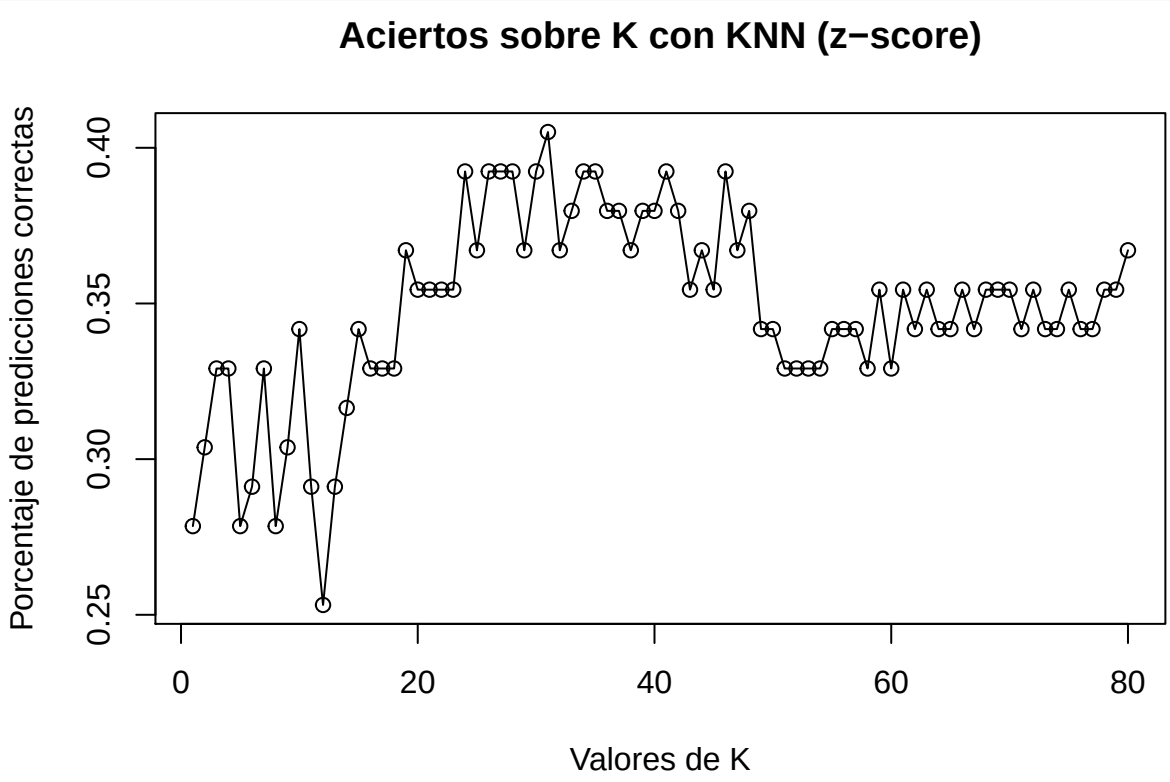
```
highest_k_mm <- which.max(acs_mm)
highest_acc_mm <- acs_mm[highest_k_mm]
```

Se puede ver que la presición para la predicción no es muy buena. Lo aciertos varian en porcentajes bajos para diferentes valores de K. Con K = 33 se da uno de los valores mas altos de accuracy (38%). Veamos qué sucede ahora si normalizamos con otra metodología.

Procesamiento con Normalización Z-score

```
# Z-score normalization
new_data <- normalize_cols(data_alc, method="zs", cols=cols_to_norm)
splits_tt <- split_train_test(new_data, percentage=.80, seed=1)
data_train <- splits_tt$train[,cols_to_norm]
data_test <- splits_tt$test[,cols_to_norm]
data_train_labels <- splits_tt$train[,11]
data_test_labels <- splits_tt$test[,11]

acs_zs <- invisible(lapply(1:80, accuracy_for_k))
plot(1:length(acs_zs), acs_zs, type='o', xlab='Valores de K', ylab='Porcentaje de predicciones correctas
```




```
highest_k_zs <- which.max(acs_zs)
highest_acc_zs <- acs_zs[highest_k_zs]
```

Se puede ver en este caso una mejora en el porcentaje de aciertos con respecto a la normalización min-max. Con K = 31 se logra un 40% de aciertos.

Como discusión final se puede hacer hipótesis del por qué un porcentaje de aciertos tan bajo. En primer lugar, al estar tratando con varias clases (5) ls posibilidades de acierto deben estar fundadas en un buen modelo de predicción, y no, tal vez, en el azar. En este sentido se podría comparar el desempeño del algoritmo

de clasificación aquí presentado con una simple elección al azar de clases. Se debe mencionar además lo naturalmente compleja que es la tarea planteada: en efecto, la predicción del consumo de alcohol de una persona nunca podrá ser predecible con excelente efectividad por ningun experto en el tema. Eso sumado a la parcialidad en las preguntas, la subjetividad inherente a las respuestas de las encuestas, y la cantidad inmensa de factores no considerados pero que pueden ser significativos en el tema tratado, resultan en un modelo con muchos puntos a mejorar.